# 小马哥的 Java 项目实战营

## SOA 项目 - 第十九节 数据架构升级

小马哥（mercyblitz）

极客大学

# 我是谁？

小马哥（mercyblitz）

- 父亲
- Java 劝退师
- Apache Dubbo PMC
- Spring Cloud Alibaba 架构师
- 《Spring Boot 编程思想》作者

极客大学

# 预习知识

- JDBC API
- Spring JDBC
- Spring Data

极客大学

# 议题

- Mybatis
- Mybatis Spring
- 问答互动

极客大学

# Mybatis

- MyBatis 配置
  - 全局XML配置文件

    MyBatis 全局XML配置文件包含影响MyBatis行为的设置和属性。

  - SQL Mapper XML配置文件

    SQL Mapper XML 配置用于映射SQL 模板语句与Java类型的配置。

  - SQL Mapper Annotation

    SQL Mapper Annotation是Java Annotation的方式替代SQL Mapper XML配置文件。

极客大学

# Mybatis

- MyBatis 配置
  - 全局XML配置文件
    - 属性（properties）
    - 设置（settings）
    - 类型别名（typeAliases）
    - 类型处理器（typeHanders）
    - 对象工厂（objectFactory）
    - 插件（plugins）
    - 环境（environments）
    - 数据库标识提供商（databaseIdProvider）
    - SQL映射文件（mappers）

极客大学

# Mybatis

- MyBatis 配置
  - 全局XML配置文件
    - 属性（properties）
      - 配置属性时作为占位符使用

```
<properties resource="org/mybatis/example/config.properties">
  <property name="username" value="dev_user"/>
  <property name="password" value="F2Fa3!33TYyg"/>
</properties>
```

```
<dataSource type="POOLED">
  <property name="driver" value="${driver}"/>
  <property name="url" value="${url}"/>
  <property name="username" value="${username}"/>
  <property name="password" value="${password}"/>
</dataSource>
```

极客大学

# Mybatis

- MyBatis 配置
  - 类型别名（typeAliases）
    - 为Java类型建立别名，一般使用更短的名称替代

```
<typeAliases>
  <typeAlias alias="Author" type="domain.blog.Author"/>
  <typeAlias alias="Blog" type="domain.blog.Blog"/>
  <typeAlias alias="Comment" type="domain.blog.Comment"/>
  <typeAlias alias="Post" type="domain.blog.Post"/>
  <typeAlias alias="Section" type="domain.blog.Section"/>
  <typeAlias alias="Tag" type="domain.blog.Tag"/>
</typeAliases>
```

极客大学

# Mybatis

- MyBatis 配置
  - 类型处理器（typeHanders）
    - 用于将预编译语句（PreparedStatement）或结果集（ResultSet）中的JDBC类型转化成Java类型

      - 如：BooleanTypeHandler 将 JDBC类型中的BOOLEAN转化成Java类型中的java.lang.Boolean 或者boolean

      - 若需要转换 Java 8 新增的Date与Time API，即JSR-310，需要再引入mybatis-typehandlers-jsr310

极客大学

# Mybatis

- MyBatis 配置
  - 对象工厂（objectFactory）
    - 用于创建结果对象实例，提供默认构造器或者执行构造参数初始化目标类型的对象。通常使用场景，不需要调整默认的实现

```
// ExampleObjectFactory.java
public class ExampleObjectFactory extends DefaultObjectFactory {
  public Object create(Class type) {
    return super.create(type);
  }
  public Object create(Class type, List<Class> constructorArgTypes, List<Object> constructorArgs) {
    return super.create(type, constructorArgTypes, constructorArgs);
  }
  public void setProperties(Properties properties) {
    super.setProperties(properties);
  }
  public <T> boolean isCollection(Class<T> type) {
    return Collection.class.isAssignableFrom(type);
  }}
```

```
<!-- mybatis-config.xml -->
<objectFactory type="org.mybatis.example.ExampleObjectFactory">
  <property name="someProperty" value="100"/>
</objectFactory>
```

极客大学

# Mybatis

- MyBatis 配置
  - 插件（plugins）
    - Mybatis提供插件的方式来拦截映射语句（mapped statement）的执行，如以下方法：
      - Executor (update, query, flushStatements, commit, rollback, getTransaction, close, isClosed)
      - ParameterHandler (getParameterObject, setParameters)
      - ResultSetHandler (handleResultSets, handleOutputParameters)
      - StatementHandler (prepare, parameterize, batch, update, query)

极客大学

# Mybatis

- MyBatis 配置
  - 插件（plugins）

```java
// ExamplePlugin.java
@Intercepts({@Signature(
  type= Executor.class,
  method = "update",
  args = {MappedStatement.class,Object.class})})
public class ExamplePlugin implements Interceptor {
  public Object intercept(Invocation invocation) throws Throwable {
    return invocation.proceed();
  }
  public Object plugin(Object target) {
    return Plugin.wrap(target, this);
  }
  public void setProperties(Properties properties) {
  }
}
```

```xml
<!-- mybatis-config.xml -->
<plugins>
  <plugin interceptor="org.mybatis.example.ExamplePlugin">
    <property name="someProperty" value="100"/>
  </plugin>
</plugins>
```

极客大学

# Mybatis

- MyBatis 配置
  - 环境（environments）
    - MyBatis 允许配置多个环境，在运行时，通过传递环境信息，切换关联的SqlSessionFactory 实例。因此，MyBatis 中的环境（environments）类似于Maven 或者 Spring 中的Profile

```
<environments default="development">
  <environment id="development">
    <transactionManager type="JDBC">
      <property name="..." value="..."/>
    </transactionManager>
    <dataSource type="POOLED">
      <property name="driver" value="${driver}"/>
      <property name="url" value="${url}"/>
      <property name="username" value="${username}"/>
      <property name="password" value="${password}"/>
    </dataSource>
  </environment>
</environments>
```

极客大学

# Mybatis

- MyBatis 配置
  - 全局XML配置文件
    - SQL映射文件（mappers）

```xml
<!-- Using classpath relative resources -->
<mappers>
  <mapper resource="org/mybatis/builder/AuthorMapper.xml"/>
  <mapper resource="org/mybatis/builder/BlogMapper.xml"/>
  <mapper resource="org/mybatis/builder/PostMapper.xml"/>
</mappers>
```

```xml
<!-- Using url fully qualified paths -->
<mappers>
  <mapper url="file:///var/mappers/AuthorMapper.xml"/>
  <mapper url="file:///var/mappers/BlogMapper.xml"/>
  <mapper url="file:///var/mappers/PostMapper.xml"/>
</mappers>
```

```xml
<!-- Using mapper interface classes -->
<mappers>
  <mapper class="org.mybatis.builder.AuthorMapper"/>
  <mapper class="org.mybatis.builder.BlogMapper"/>
  <mapper class="org.mybatis.builder.PostMapper"/>
</mappers>
```

```xml
<!-- Register all interfaces in a package as mappers -->
<mappers>
  <package name="org.mybatis.builder"/>
</mappers>
```

极客大学

# Mybatis

- MyBatis 配置
  - XML定义
    - 文档类型约束方式
      - DTD：Document Type Definition
      - http://mybatis.org/dtd/mybatis-3-config.dtd
    - API接口
      - org.apache.ibatis.session.Configuration

极客大学

# Mybatis

- MyBatis 配置
  - 属性（properties）
    - 配置内容

```
<properties resource="org/mybatis/example/config.properties">
  <property name="username" value="dev_user"/>
  <property name="password" value="F2Fa3!33TYyg"/>
</properties>
```

    - 组装API接口
      - org.apache.ibatis.session.Configuration#variables
    - 填充配置
      - org.apache.ibatis.builder.xml.XMLConfigBuilder(XPathParser,String, Properties)

极客大学

# Mybatis

- MyBatis 配置
  - 设置（settings）

```xml
<settings>
  <setting name="cacheEnabled" value="true"/>
  <setting name="lazyLoadingEnabled" value="true"/>
  <setting name="multipleResultSetsEnabled" value="true"/>
  <setting name="useColumnLabel" value="true"/>
  <setting name="useGeneratedKeys" value="false"/>
  <setting name="autoMappingBehavior" value="PARTIAL"/>
  <setting name="autoMappingUnknownColumnBehavior" value="WARNING"/>
  <setting name="defaultExecutorType" value="SIMPLE"/>
  <setting name="defaultStatementTimeout" value="25"/>
  <setting name="defaultFetchSize" value="100"/>
  <setting name="safeRowBoundsEnabled" value="false"/>
  <setting name="mapUnderscoreToCamelCase" value="false"/>
  <setting name="localCacheScope" value="SESSION"/>
  <setting name="jdbcTypeForNull" value="OTHER"/>
  <setting name="lazyLoadTriggerMethods" value="equals,clone,hashCode,toString"/>
</settings>
```

极客大学

# Mybatis

- MyBatis 配置
  - 设置（settings）
    - XML声明
      - <settings> 元素
        - <setting> 元素
    - 组装API接口
      - org.apache.ibatis.session.Configuration#setXXX(*)
    - 填充配置
      - org.apache.ibatis.builder.xml. XMLConfigBuilder#settingsElement(Properties)

极客大学

# Mybatis

- MyBatis 配置
  - 类型别名（typeAliases）
    - XML声明
      - <typeAliases> 元素
        - <typeAliase> 元素
    - 组装API接口
      - org.apache.ibatis.session.Configuration#typeAliasRegistry
    - API定义
      - org.apache.ibatis.type.TypeAliasRegistry
    - 填充配置
      - org.apache.ibatis.builder.xml.XMLConfigBuilder#typeAliasesElement(XNode)

极客大学

# Mybatis

- MyBatis 核心API
  - org.apache.ibatis.session.SqlSessionFactoryBuilder
    - SqlSessionFactory 构建器，创建 SqlSessionFactory 实例。通过重载方法build，控制实例行为，其中方法参数如下：
      - MyBatis全局配置流（java.io.InputStream、java.io.Reader）
      - Mybatis环境名称（environment）
      - Mybatis属性（java.util.Properties）
    - 相关API
      - 配置构建器：org.apache.ibatis.builder.xml.XMLConfigBuilder
      - MyBatis配置：org.apache.ibatis.session.Configuration
      - MyBatis环境：org.apache.ibatis.mapping.Environment

极客大学

# Mybatis

- MyBatis 核心API
  - org.apache.ibatis.session.SqlSessionFactory
    - SqlSession 工厂类，创建 SqlSession 实例，通过重载方法openSession，控制实例特性，其中方法参数如下：
      - 是否需要自动提交
      - JDBC 数据库连接（java.sql.Connection）
      - Mybatis SQL语句执行器类型（org.apache.ibatis.session.ExecutorType）
      - Mybatis 事务隔离级别（org.apache.ibatis.session.TransactionIsolationLevel）

极客大学

# Mybatis

- MyBatis 核心API
  - org.apache.ibatis.session.SqlSession
    - MyBatis SQL 会话对象，类似于JDBC Connection
      - 职责
        - 封装java.sql.Connection
        - 屏蔽java.sql.Statement（以及派生接口）的细节
        - 映射java.sql.ResultSet 到Java类型
        - 事务控制
        - 缓存
        - 代理映射（Mapper）

极客大学

# THANKS!